SYSTEM AND METHOD FOR AUTOMATIC TRANSMISSION OF AUDIBLE ON-LINE ANALYTICAL PROCESSING SYSTEM REPORT OUTPUT

Related Applications

This application claims priority from U.S. Provisional Application Nos. 60/126,055 entitled "System and Method for Automatic Transmission of On-Line Analytical Processing System Report Output" and 60/153,222 entitled "System and Method For the Creation and Automatic Deployment of Personalized, Dynamic and Interactive Voice Services" is a continuation-in-part of U.S. Patent Application No. 09/343,561, entitled "System and Method for Adaptable Automatic Transmission of OLAP Reports to Output Devices," filed on June 30, 1999.

Field of the Invention

This invention relates to a system and method for creation and automatic deployment of audible personalized, dynamic and interactive services, including information derived from on-line analytical processing (OLAP) systems, to a two-way communication device, including electronic mail, two-way pagers, phones, personal digital assistants, and telephones, based on subscriber-specified criteria.

Background of the Invention

The ability to act quickly and decisively in today's increasingly competitive marketplace is critical to the success of any organization. The volume of data that is available to organizations is rapidly increasing and frequently overwhelming. The availability of large volumes of data presents various challenges. One challenge is to

avoid inundating an individual with unnecessary information. Another challenge is to ensure all relevant information is available in a timely manner.

Although some push technologies have been developed for automatically delivering content to users, most systems simply "dump" information about a particular subject without regard to users' particular preferences or interests. Some such technologies are available on the World Wide Web and the Internet.

The World Wide Web and the Internet have provided an avenue for information delivery, but current Web-based systems still fail to adequately deliver the right information at the right time. One of the major problems with the World Wide Web is the requirement to utilize a computer and web-browser to access its contents. Although penetration of computers throughout the world has increased, that penetration is far from making information readily available to everyone wherever they happen to be.

Moreover, most computer users connect to the Web through a land line. Most users therefore do not have access to Web content when they are away from a land line. Although technology is being developed to enable World Wide Web access through other mediums, such as web-enabled personal digital assistants, for example, such technology require users to purchase new equipment to access this technology. Given the sparse penetration of personal digital assistants already, this technology does not satisfy the need for delivery of timely information. Also, these systems require the user to initiate the connection to the Internet through the PDA.

These and other drawbacks exist with current systems.

Summary of the Invention

An object of the invention is to overcome these and other drawbacks in existing systems.

Another object of the present invention is to provide a system that automatically broadcasts audible messages to subscribers based on criteria established by the subscriber when those criteria are determined to be satisfied by an on-line analytical processing system.

Another object of the present invention is to provide a system and method for automatically broadcasting messages to two-way communications devices.

These and other objects are realized by a system and method according to the present invention as described below. Such a system and method comprises a broadcast module that connects to an on-line analytical processing (OLAP) system comprising a server system for accessing information in one or more data warehouses to perform report analysis. The broadcast module may enable the defining of a service. A "service" as used herein should be understood to include one or more reports that are scheduled to be run against one or more data warehouses, relational databases, files in a directory, information from web or file transfer protocol sites, or information provided by a custom module, by a server system. These services may be subscribed to by users or user devices to enable the broadcast module to determine who should receive the results of a service.

The broadcast module enables the creation of a service, the running of a service at based on the occurrence of a predetermined condition (such as a scheduled time, a trigger condition or an initiation request), scheduling of the service, subscription of users to the

defined services, generation of reports for the service, formatting of outputs of the service and broadcasting of audible messages based on the output for the service, among other functions. The broadcast module may be connected to a plurality of two-way communication devices including a personal digital assistant (PDA), pager, mobile phone, telephone, electronic mail, and web page interface.

According to one embodiment of the present invention, a system for automatically generating output from an on-line analytical processing system based on scheduled services specified by subscribers of the system is provided. The system processes scheduled services in an on-line analytical processing system with each service comprising at least one query to be performed by the on-line analytical processing system. The system then automatically forwards output from the services to one or more subscriber output devices specified for that service. Users may define new services, including the schedule of the services and the type, such as alert services or scheduled services, and may also subscribe to the services provided by the system. If an alert service is processed, the system may forward output only when one or more alert criteria are satisfied. Subscribers may be specified by a dynamic recipient list that is resolved each time the service is processed to determine recipients of the service output. A dynamic recipient list may be, for example, a list that determines the recipients of a service based on dynamically resolved criteria. For example, a bank may generate a list every month that identifies customers who have an account balance greater than $100,000 and who have not made a transaction within the last three months. The output devices the

system may forward output to may comprise electronic mailbox, facsimile, printer, mobile phone, telephone, pager, PDA or web pages.

Additionally, audible content may be provided by a voice-enabled system that broadcasts voice content to users and even enables users to respond through a two-way communications device. In such a system, a voice service comprises at least six characteristics: service type, content type, schedule, content definition, personalization settings and error handling settings. Once a voice service is set-up, a user can subscribe to the voice service. Based upon the occurrence of a predetermined event (for example, a schedule or a trigger event) a telecast may be executed. Execution of a telecast preferably comprises the steps of generating voice service content, creating telecast call structure, sending telecast to the service queue, processing a call request and writing subscriber responses. Preferably during each telecast, a synthesized, natural-sounding voice greets the recipient by name, identifies itself and provides information relevant to the subscriber. The system then either prompts the person for a response or presents a choice of menu options that lead to more dialog.

The system creates fully personalized calls by generating an XML-based document (Active Voice Page) for every message that goes out. This document then determines in real-time the individual call flow. Call content varies depending on the user and the responses that are entered. By simply using the telephone keypad (or other input mechanism) users can control the entire call flow, select given options, enter information at user prompts and conduct transactions. Additionally, the system can collect user inputs and provide them to external applications, databases and transaction systems.

The voice services are based on real-time, text-to-speech conversion, to allow for truly personalized messaging. Unlike other telephony applications such as phone-banking or phone-based trading services that traditionally use static pre-recorded files, Text-to-speech does not impose any limitations on the content of the call. The system speaks customer names, product brands and other terms. Further, it leverages specific algorithms to resolve special content such as numbers, phone numbers or dates etc. In certain cases, it may be beneficial to include pre-recorded dialog components and other audio files (sound effects, music, testimonials, etc.). Thus, the system enables blending of static and dynamic content in various part of the message.

Another embodiment of the invention relates to a system and method for creation and automatic deployment of personalized, dynamic and interactive voice services, including information derived from on-line analytical processing (OLAP) systems and other data repositories. The system and method enables the ability to capture user selections to facilitate closed-loop transaction processing and processing of other requests. One aspect of the invention relates to an interactive voice broadcasting system and method that enables analytical reporting and advanced transactional services via the telephone or other voice-enabled terminal device. One advantage of the invention is that a voice service may leverage the power of OLAP or other data repository systems and provide critical information to the user, in a timely fashion, by phone. Another advantage of this method and system is that it provides a user with the opportunity to immediately act upon information received during a interactive voice broadcast.

A voice service is created and can have many users subscribed to the voice service. Each user can specify personal preferences for the content and presentation of the contents for a voice service. The specification of the elements of a voice service may be done using a set of interfaces (such as GUIs) that take the form of a voice service wizard.

A voice service includes one or more Dialog elements. Dialog elements may include one or more of Speech elements, Input elements and Error elements. An Input element may include a Prompt element and/or an Option element. An Input element enables the system to request input from the user, capture the input and direct the call flow based on the user's input. An Option element associates a key (e.g., on a telephone touch pad dial) with a destination Dialog that is executed when that number is pressed by a user during an interactive voice broadcast. A Prompt requests a user to enter numeric or other information. An Input element may enable a user to request, during an interactive voice broadcast, a transaction, a service or other requests. The term transactions, services and requests are to be interpreted broadly.

According to one embodiment, the user's responses to Input elements are stored during an interactive voice broadcast and, during or after the voice broadcast, the stored information is processed by the system or is passed to another system or application for processing. The transaction (or other request) processing can be accomplished either in real-time, during the voice broadcast, or after the interactive voice broadcast is completed. The results or confirmation of a transaction or other request can be provided to the user during the call or subsequently.

Once a voice service is created, the system monitors predetermined conditions to determine when the voice service should be executed. Each voice service is executed when one or more predetermined conditions are met as specified during creation of the voice service. For example, a voice service may be executed according to a predetermined schedule (time-based) or based on a triggering event (e.g. one or more conditions are met based on the output of an OLAP or other report).

When the predetermined condition is satisfied, the voice service is executed. Executing a voice service, includes the steps of generating the content specified by the voice service and the user preferences. Some users may have identical personalization options and, thus, a single call structure may be generated for a group of users with identical personalization options. The content of the voice service includes the information that is to delivered to users of that voice service, and the Input to be requested from the user, among other things. The content may include, for example, static text messages, dynamic content (e.g. text based on information output from an OLAP report, other database or other sources) or blended text (e.g. static text combined with dynamic content).

This and other content are formatted in an Active Voice Page (AVP). An AVP contains the call structure and data. The AVP contains data at various hierarchical levels that are defined by the Dialog elements defined for each voice service. The active voice pages are used to help govern the interaction between the call server and the user during an IVB. According to one embodiment, the content is formatted, into an AVP e.g., using XSL stylesheets so the AVP is in an XML-based language. According to one

embodiment, the XML-based language used is a novel language referred to as TML (discussed below). The AVP is sent to a call server along with style properties for each user. The style properties of a user help determine the behavior of the call server during an interactive voice broadcast. A unique AVP is generated for each user scheduled to receive a voice service.

When a user is called by the call server, information is passed through a T-T-S engine and delivered to the user through a voice-enabled terminal device. Preferably, the structure of each call is dynamic, driven by current data values and is personalized based on a user profile established during subscription to a voice service. During a typical interactive voice broadcast, a synthesized, natural sounding voice greets the recipient by name, identifies itself, provides information relevant to the user and enables a user to provide input back to the system.

An IVB is a voice-enabled interaction with a user having a dynamic structure controlled by the AVP for the particular user. The IVB may be delivered using real-time, on-the-fly speech generation. During an IVB, information is exchanged between the call server and a user according to the AVP. The system executes dialogs by reading messages to the user and, eliciting input from the user. For example, the user may press buttons on a telephone touch pad dial to select an option or to provide numeric or alphanumeric input. Each response provided by a user may transfer control of the IVB to a different part of the AVP.

During or after the IVB, the user's responses may be processed by the system or other applications. The AVP may contain pointers to other applications and embedded

statements such that when a user exercises an option, the system performs a requested operation and returns the results to the user during the IVB. For example, by exercising an option, a user may request that a real-time database query be performed. When the user selects such an option, control is shifted to a portion of the AVP that contains an embedded SQL statement that is made against a database.

When a user has worked through selected dialogs of the AVP, the IVB is terminated. That is, a user likely will not work through all of the available dialogs during an IVB. Rather, the user's inputs and option selections determine which the available dialogs are encountered during any given IVB.

Other objects and advantages of the present invention will be apparent to one of ordinary skill in the art upon reviewing the detailed description of the present invention.

Brief Description of the Drawings

Fig. 1a is a flow chart of a method in accordance with an embodiment of the present invention.

Fig. 1b is a flow chart indicating a method of generating a voice service according to one embodiment of the present invention.

Fig. 1c is a flow chart indicating a method for interactive voice broadcasting according to an embodiment of the present invention.

Fig. 2 is a flow chart indicating a sequence of an interactive voice broadcast according to one embodiment of the present invention.

Fig. 3a is a schematic block diagram of a system in accordance with an embodiment of the present invention.

Fig. 3b is a schematic block diagram of an intelligence server according to an embodiment of the present invention.

Fig. 3c is a schematic block diagram of call server according to an embodiment of the present invention.

Fig. 4 is a schematic block diagram of a commercial transaction processing system according to an embodiment of the present invention.

Fig. 5 is a flow chart of a method of using a voice service bureau according to an embodiment of the present invention.

Fig. 6a is a schematic block diagram of a voice service system incorporating a voice service bureau according to one embodiment of the present invention.

Fig. 6b is block diagram of a primary voice bureau according to one embodiment of the present invention.

Fig. 6c is a block diagram of a backup voice bureau according to another embodiment of the present invention.

Fig. 7 is a flow chart illustrating a method for integrating inbound and outbound voice services.

Fig. 8 is a block diagram of a call server configured to provide integrated inbound and outbound voice services.

Fig. 9 is a schematic block diagram of a system in accordance with an embodiment of the present invention.

Fig. 10 is a schematic block diagram of a method for automatic transmission of OLAP report information.

Fig. 11 is a schematic block diagram of a method for creating a service according to an embodiment of the present invention.

Fig. 12 is a schematic block diagram of an overall system in accordance with an embodiment of the present invention.

Detailed Description of the Preferred Embodiments

The invention of the present invention enables automatic report generation from an OLAP system, or any other database structure, to two-way communication devices and also enables audible output, such as voice content, to those devices.

According to one embodiment, an overall system may be described with reference to Figs. 9-12. An additional embodiment is also provided that specifically enables voice output. These embodiments illustrate a system for audible content delivery and delivery of content to two-way communications devices.

According to one embodiment of the present invention, a system is provided for automatic transmission of OLAP report output to one or more of a plurality of user output devices. Fig. 9 depicts an embodiment of a system 2100 according to the present invention. System 2100 may comprise a data warehouse 2012, a server system 2014, a broadcast module 2020, an object creation module 2024, an agent module 2028, and one or more user devices 2040. User devices 2040 may comprise a facsimile 2040a, pager 2040b, mobile telephone 2040c, electronic mail 2040d, and web page output 2040e.

Broadcast module 2020 may comprise a module that broadcasts personalized information derived from the OLAP system (*e.g.*, data warehouse 2012 and sever system 2014) to users via one or more user devices 2040 such as electronic mail, facsimile,

printer, pager, mobile phone, telephone, PDA, and multiple other types of user information devices. Broadcast module 2020 enables users to define services (*e.g.*, queries and reports) that are to be run against an OLAP system such as server system 2014 and data warehouse 2012 based on a predetermined schedule. A "service" as used herein should be understood to include one or more reports that are scheduled to be run against data warehouse 2012 by server system 2014. Broadcast module 2020 also enables users on the system to subscribe to one or more services and then broadcast module 2020 outputs the results of these services to subscribers according to criteria established by the subscribers.

Broadcast module 2020 may generate output to a plurality of different types of output devices. Broadcast module 2020 interacts with multiple types of output modules. The output modules may include Mail/Messaging Applications Programming Interface (MAPI) (*e.g.*, for Rich Text Format (RTF) electronic mail), Simple Mail Transfer Protocol (SMTP) (*e.g.*, for markup language documents), facsimiles, pager service provider application program interfaces, mobile phone networks, wireless access protocols, Object-Oriented Database Connectivity (ODBC), HyperText Transfer Protocol (HTTP) posts, flat files, printers, or output modules customized by a user.

In one embodiment, a styles module may be provided. The styles module may enable system administrators or users to specify the formatting requirements of a user device 2040 so that broadcast module 2020 may generate output to that device in a manner that users can understand. Each style may be designed to suit particular users' needs and may be generally provided for a particular brand or type of device. For

example, different styles may be defined for each different type of pager, facsimile output devices, and different types of electronic mail systems. Styles may thus be specified for various output devices including electronic mail, pagers, facsimiles and mobile telephones.

Rather than hard-coding the format of each possible output device, the present invention provides a plurality of styles parameters that may be modified for each particular style. This may be accomplished using the styles module. The parameters for each particular user device 2040 may then be provided and broadcast module 2020 may be able to output information to user device 2040 based on the values provided for the parameters provided. In this way, the system is configurable to new devices and different device formats by specifying values for the parameters that satisfy that device.

Specifically, the styles module formats may include a plurality of parameters including general parameters, enclosure handling parameters, text grid parameters, and advanced formatting parameters. Each group of parameters may comprise a plurality of properties. The general parameters may comprise the following properties and sample value ranges: name (any string up to 50 characters), description (text grid up to 5000 characters), display width (*e.g.*, total number of characters available on a row of the display) (1-9999 or unlimited), segment size (*e.g.*, maximum number of characters that can be in a single message) (1-9999 or unlimited), URLs allowed (*e.g.*, whether URL outputs are allowed in the message) (True/False, *i.e.*, T/F), URL unavailable message (*e.g.*, whether message which is inserted in place of the URL tag when the URL cannot be created because of personalization issues) (text string up to 248 characters), body element

separator (*e.g.*, character string which separates an alert element/value from the next) (any valid character plus carriage return for the device), supports Rich Text Format (RTF) (*e.g.*, dictates whether RTF codes are to be stripped from the body of the message before it is sent) (T/F), default style (*e.g.*, determine whether a style is the default style) (Y/N), truncate alerts (*e.g.*, alert rows that do not completely fit in a segment entirely moved to the next segment) (T/F), fill row (*e.g.*, appends spaces to an alert or grid row to force the next row onto a new line) (T/F), body element separator (*e.g.*, string which separates an alert element/value from the next) (valid character string up to 20 characters or carriage return), and truncation character (*e.g.*, user specified character that is used to segment messages) (valid characters).

Enclosure handling parameters may comprise the following properties and valid values for each: enclosure flag (*e.g.*, indicates whether enclosures should be sent) (T/F), convert enclosure (*e.g.*, indicates that if convert enclosures is true, the enclosures generated by the broadcast module should be put into the message itself (for example, all Excel grids should be converted to text grids or discarded if in a non-standard format, such as a Word document)) (T/F), enclosure removed message (ERM) (*e.g.*, if enclosure flag is not true, this ERM is placed into the outgoing message in place of the enclosure) (valid text string up to 248 characters), attachment placement (*e.g.*, places attachments in a specified location instead of mail server default) (end, offset by user specified number of carriage returns), and enclose facsimile RTF (*e.g.*, converts RTF to attachment for broadcast via facsimile) (T/F).

Text grid parameters may comprise the following properties and sample values: column pad (*e.g.*, character placed immediately after a column element's data so that all columns line up properly) (null, space, or valid character), column separator (*e.g.*, character used to delimit a column) (null, space, tab or valid character), grid space (*e.g.*, character used to replace the space character everywhere in a grid) (null or any valid character), column overflow character (*e.g.*, character separating columns when displayed in overflow mode) (null, space, tab or valid character), and truncate rows (*e.g.*, grid rows that do not completely fit in a segment may be entirely moved to the next segment) (T/F).

Advanced formatting parameters may include the following properties and sample values: subject line length (*e.g.*, number of characters on the subject line) (0-250), subject pad (*e.g.*, character used to stuff the subject line to its maximum length) (null, space or valid character), subject space (*e.g.*, character used to replace the space character everywhere in the subject) (null, or valid character), truncation string (*e.g.*, indicates that some data has been cut off and may appear at the end of every segment except the last one) (null, text string up to 20 characters), maximum segments (*e.g.*, indicates the maximum number of chunks in which the data can be sent) (*e.g.*, 0-9999 or unlimited), CR replacement (*e.g.*, replaces each vbCrLf to force carriage returns on the output medium and may be used as a row separator for a text grid) (carriage return, null, space, tab or valid character), body subject (*e.g.*, places message body in subject (for some mobile phones), message subject in body (*e.g.*, for pagers that do not support subject lines)) (T/F), segment prefix (*e.g.*, string attached to beginning of each segment) (null, valid text string up to 20 characters, carriage return, space, segment number, date/time

sent), and message body in subject (*e.g.*, places message in subject line (for some mobile phones)) (T/F).

The value of the properties described above may be used by the styles module to format the output to be sent to the output device defined for that style. The segment module may perform the following steps based on the value of the properties in each of these parameters for the particular style to receive the output message.

Padding and formatting of the message may be performed as specified in the properties and then character removal and replacement may be performed on the message body generated by broadcast module 2020 from the server module. The message size may then be compared to the allowable size to determine whether segmentation is necessary.

If so, the following steps may be performed for segmentation. First, the last possible break character is determined (this character may be the position given by the start of the segment plus the segment size minus the length of the truncation string). The module then looks for the first character before the last possible character that makes the most sense for breaking.

To determine the character for breaking, the module first determines what is the newline indicator in this message. If carriage returns have been replaced, then the carriage return replacement character is the newline indicator and, if not, then the carriage return is the newline indicator.

If the newline indicator immediately precedes the last possible break character, then the newline indicator is the place for the break-in segments. If not, then the styles

module defines a search zone of characters up to and including the last possible break character. The number of characters in the search zone may be the minimum of the display width of the style or one third of the allowable length of the segment (segment size minus length of truncation string). If a newline indicator is in the search zone, then the styles module takes the last newline indicator in the search zone and places the break immediately before it. If no newline indicator is found, then the styles module looks for spaces or tabs. The first space or tab preceding the last possible break character is then selected as the break point. If no such space or tab is in the search zone, then the break point is between the last possible break character and the next character. The start of the next segment is the first character after the break point that is not a space, tab, or part of a newline indicator.

A truncation string may be placed in each segment to indicate that information has been truncated due to limits on the number of segments that the output is to receive. That way, the user may be informed that additional information was not sent that may be important and that the user may desire to check the full report or request that the report be resubmitted with all of the information (*i.e.*, an override on the limit of segments).

If the subject-in-body property is true has been selected, then the module concatenates the subject and body text of the message with a space in between and places this in the body of the message. If the body-in-subject property is true, then the module concatenates the subject and body of the message with a space in between, and places this in the subject line. The body may be truncated to fit the subject size limitations.

The styles module may operate with the broadcast module 2020 in an environment as depicted in Fig. 9. In such a system, data warehouse 2012 may comprise any data warehouse or data mart as is known in the art, including a relational database management system ("RDBMS"), a multidimensional database management system ("MDDBMS") or a hybrid system. Server system 2014 may comprise an OLAP server system for accessing and managing data stored in data warehouse 2012. Server system 2014 may comprise a ROLAP engine, MOLAP engine or a HOLAP engine according to different embodiments. Specifically, server system 2014 may comprise a multithreaded server for performing analysis directly against data warehouse 2012. One embodiment of server system 2014 may comprise a ROLAP engine known as DSS Server™ offered by MicroStrategy. Accordingly, data warehouse 2012 and server system 2014 comprise an OLAP system that connects to broadcast module 2020 for broadcast of user-specified reports from data maintained by data warehouse 2012.

Broadcast module 2020 may also be connected to an agent module 2028 which may also be connected to server system 2014. Agent module 2028 may be provided to define reports and queries that may be selected as part of one or more services by broadcast module 2020. Agent module 2028 may be used to define queries to be performed against the data contained in data warehouse 2012 using components, templates, filters, reports, agents, etc. Components may include dimensions, attributes, attribute elements, and metrics – in other words, the building blocks for templates, filters, and reports. Templates generally define a report format and specify the attributes, dimensions, metrics, and display properties comprising a report. Filters generally qualify

report content and identify a subset of data warehouse 2012 to be included in a report. For example, filters may be used with set math, multidimensional qualifications, and metric qualifications. Using set math, users can define and embed any set of limiting criteria (*e.g.*, union, intersect, exclude). Multidimensional qualifications enable users to indicate general subject areas or perspectives on data (*e.g.*, time, geography, product). Metric qualifications may be used to compute mathematical calculations of various numerical data (*e.g.*, total sales, profit, cost, percent change, profit). Metrics may be displayed in a variety of formats (*e.g.*, percentages, currency, fonts indicating predetermined values). Reports are generally understood to be a data analysis created by combining a template (the format) with a filter (the content). Agents may be a group of reports cached on a time- or event-based schedule for rapid retrieval and batch processing. According to one embodiment of the invention, agent module 2028 may comprise a software package known as DSS Agent™ offered by MicroStrategy.

Agent module 2028 may operate on any user system 2026 including personal computers, network workstations, laptop computers or any other electronic device connected to server system 2014 or may comprise an object connected to broadcast module 2020.

Broadcast module 2020 therefore, cooperates with server system 2014 and agent module 2028 to send personalized information to users at predefined intervals or when criteria specified in reports defined through either broadcast module 2020 or agent module 2028 exceed predefined thresholds. To provide this functionality, broadcast module 2020 enables users of the system to create services that run against the OLAP

system to generate information and subscriptions that specify the recipients of the information derived from a service. A service may comprise one or more reports that are processed by the OLAP system and may be a specific report, series of reports or elements within a report. Also, subscribers may include users, groups of users or only specific user devices 2040 for a particular user. Services may be based on predefined reports from broadcast module 2020 or agent module 2028 or may be based on filter/template combinations set up through broadcast module 2020 and/or agent module 2028.

Once services have been defined and subscribers to that services are established, broadcast module 2020 continually monitors the schedules for the services, runs the scheduled reports, and automatically generates outputs where conditions specified in the service are satisfied using push technology. Outputs from broadcast module 2020 may be personalized to subscriber demands and/or formatted to meet a subscriber's user device requirements to ensure that users see only that portion of a report that is relative to that user and in a manner that is most useful for the user. Accordingly, a user can thus have up-to-date information about the contents of data warehouse 2012 without having to submit a query or log-in to a software module on the user system.

To provide the functionality described above, broadcast module 2020 may comprise a plurality of modules that perform certain functions. Although described as separate modules, it should be understood that such modules may be combined or separated further. In an embodiment of the present invention, as depicted in Fig. 9, broadcast module 2020 may comprise a service definition module 2042, a service schedule module 2044, a service generation module 2046, a service format module 2048,

a personalization module 2050, a subscription interface module 2052, and a service administration module 2054.

Service definition module 2042 of broadcast module 2020 may comprise a module for enabling a user to create or modify a service. In an embodiment, the services may be defined based on reports or workbooks specified in agent module 2028. Users may then subscribe to services defined in service definition module to enable broadcast module 2020 to determine who should receive the results of a service.

At least two types of services may be provided – scheduled services and alert services. A schedule service may comprise a service that generates information to subscribers at a given time interval. An alert service may comprise a service that provides information to all subscribers if an alert condition is true.

Service schedule module 2044 may provide the functionality to enable selection of when a service should be run. Service schedule module 2044 may enable a user, administrator or other person having access thereto to specify the frequency that the service should be performed. The schedule may be based on an interval (such as every several hours, days, weeks, months, years, etc.) or on one or more specified days (such as March 15th and September 15th). Other methods of scheduling events to be processed may also be used.

Service generation module 2046 may comprise a module for following a schedule set by service schedule module 2044 and completing the operation specified in service definition module 2042. For example, if a service were specified to run the monthly sales totals for the Midwest region of a company every weekend and generate an alert to the

supervisor on Monday morning if sales drop below 5%, then service generation module 2046 would be responsible to monitor the schedule of this service to ensure that the report contained therein was processed over the weekend and then generate an alert report if the criteria set in the service is satisfied. To monitor the schedule of all services specified by broadcast module 2020, service generation module 2046 may operate constantly to ensure that every scheduled service is completed.

Service format module 2048 may be responsible for taking the results of a service and formatting it to a proper format corresponding to each of the subscribers of a particular service. Service format module 2048 may be responsible for formatting service results for generation to user devices 2040a-2040e.

Personalization module 2050 may be provided to enable subscribers to specify the content for a service in which they are interested. Users may input personalized choices for personalization module 2050 through subscription interface module 2052 by selecting personalization filters from filters available in the service. Personalization module 2050 captures the criteria selected by the user and creates a subscription based on the selected criteria which may be multidimensional based on the data structure in the data warehouse, relational database, etc. Because personalization module 2050 enables subscribers to specify the content of a service, this reduces the amount of data output to a subscriber by providing the subscriber with data that the subscriber is interested in.

Personalization may also be set in an address book module maintained by broadcast module 2020. The address book may comprise an entry for each subscriber of any service on the system. That subscriber may define global personalization filters to be

applied to all services to which the subscriber applies rather than providing personalization on only a service by service basis. For a chosen user, personalization may also be set on a project level basis. For example, a project may comprise multiple reports. Each report within a project may have different personalized filters applied according to user desires. Personalization module 2050, however, also enables subscribers to personalize the entire rather than doing so for each. For example, subscribers may assign particular operations to be performed for each of the reports within a project. This enables subscribers to personalize multiple reports simultaneously.

Additionally, subscribers may personalize style parameters using, for example, personalization module 2050. Styles may be used to tailor a display format of a report to a particular device (*e.g.*, pager, electronic mail, facsimile). Styles may be designed according to the needs of each subscriber depending on the characteristics and properties of a recipient. For example, a user may desire to have pages generated from the report sent in a particular format and may set up that format using styles.

Further, in the address book, the data warehouse login and password may also be stored to enable access to the data warehouse. Each subscriber may have multiple addresses in the address book, each address assigned to a different output device. Thereby, a subscriber may have multiple addresses, each of which may be assigned to receive different services based on different filters, etc. This provides for the ultimate in customization for receipt of information. For example, a user may desire to get stock information via pager but sales information via e-mail. By setting up two separate

addresses and applying different services and/or filters, that may be accomplished according to the present invention.

Subscription interface module 2052 may be provided to enable users or administrators of the system to monitor and update subscriptions to various services provided by broadcast module 2020. Service administration module 2054 may be provided to provide administrative functions to monitor a queue to schedule services and to provide throughput of services to ensure efficient completion of those services by broadcast module 2020.

Subscription interface module 2052 may be used to create a subscriber list by adding one or more subscribers to a service. Users or system administrators having access to broadcast module 2020 may add multiple types of subscribers to a service such as a subscriber from either a static recipient list (SRL) (*e.g.*, addresses and groups) or a dynamic recipient list (DRL) (described in further detail below). The subscribers may be identified, for example, individually, in groups, or as dynamic subscribers in a DRL. Subscription interface module 2052 permits a user to specify particular criteria (*e.g.*, filters, metrics, etc.) by accessing data warehouse 2012 and providing the user with a list of available filters, metrics, etc. The user may then select the criteria desired to be used for the service.

A SRL is a list of manually entered names of subscribers of a particular service. The list may be entered using service administration module 2054 or subscription interface module 2052. SRL entries may be personalized such that for any service, a personalization filter (other than a default filter) may be specified. A SRL enables

different personalizations to apply for a login alias as well. For example, a login alias may be created using personalization module 2050. Personalization module 2050 enables subscribers to set preferred formats, arrangements, etc. for displaying service content. The login alias may be used to determine a subscriber's preferences and generate service content according to the subscriber's preferences when generating service content for a particular subscriber.

A DRL may be a report which returns lists of valid user names based on predetermined criteria that are applied to the contents of a database such as data warehouse 2012. Providing a DRL as a report enables the DRL to incorporate any filtering criteria desired, thereby allowing a list of subscribers to be derived by an application of a filter to the data in data warehouse 2012. In this manner, subscribers of a service may be altered simply by changing the filter criteria so that different user names are returned for the DRL. Similarly, subscription lists may be changed by manipulating the filter without requiring interaction with service administration module 2054. Additionally, categorization of each subscriber may be performed in numerous ways. For example, subscribers may be grouped via agent filters. In one specific embodiment, a DRL is created using DSS Agent™ offered by Microstrategy.

Service administration module 2054 enables monitoring of reports (*e.g.*, ability to see who is using system, what reports they are generating, etc.), scheduling of reports, address book and dynamic recipient list maintenance, and subscriber management. Subscriber management involves enabling system administrators to review, access, and generate information about subscribers to the system through the maintenance of detailed

subscriber lists including the DRL's and SRL's. This list may track information on which subscribers subscribe to which services and vise-versa.

A method 2200 of operation of system 2100 is provided in Fig. 10. Method 2200 comprises several steps for generating information to a plurality of user systems using "push" technology. In step 2100, one or more services are defined by users or system administrators for broadcast module 2020 to monitor, as described in more detail below with respect to Fig. 11, such as through service definition module 2042. In step 2102, subscribers for each of these various services are provided, such as through subscription interface module 2052. In step 2104, the system monitors and processes services according to their defined schedules. Step 2104 may be performed by service schedule module 2044, and/or service generation module 2046, for example.

In step 2106, the system determines whether an alert criteria has been met or if a scheduled service has been completed, such as through service generation module 2046. If an alert criteria has not been satisfied or a scheduled service has not completed, the system continues to monitor and process services. If an alert condition has been met, or if a scheduled service has been completed, in step 2108, the system, such as through service generation module 2046, builds the service output and the subscription list for that particular service. Building the subscription list for a service may involve using a recipient list resolution method. For example, a recipient list resolution (RLR) may be used to build a list of all of the subscribers to a service in step 2108. This may be performed by resolving and merging all DRLs with all SRLs for a given service. All DRLs are generated and the resulting list is merged with the SRL. Typically, there is only

one SRL (although additional SRLs may be used) and none to numerous DRLs per service. The list that results from merging all of the DRLs and SRLs produces a list which consolidates all subscribers of a given service.

Next, in step 2110, the system, such as through personalization module 2050, applies personalization filters to services that are scheduled to be output to the subscribers. Personalization filters may modify the output of a service according to the subscriber's desired criteria. The personalized outputs may then be formatted for the user device 2040 selected by the user for output. Additionally, personalization module 2050 may also be used to personalize the contents of one or more services as described above. In step 2114, broadcast module 2020 broadcasts the formatted and personalized services to subscribers at user devices 2040a-2040e.

As described above, step 2100 defines the service or services to be monitored by broadcast module 2020. Fig. 11 depicts a method 2210 according to one embodiment of the present invention for performing step 2100. According to one embodiment, in step 2116, a user may name and provide a description of the service or services to be monitored. By providing a name and description, users may be able to uniquely identify the services from an object browser or in a service queue.

Next, in step 2118, the user selects the type for the service. As described above, at least two types of services may be provided. A first type, a scheduled service, is a service that is run according to a predetermined schedule and output is generated each time the service is run. An alert service is one that is run periodically as well, however, output is only generated when certain alert criteria is satisfied. If an alert service is

selected by the user, the user may then specify a report or a template/filter combination upon which the alert is based. According to one embodiment, reports and template/filter combinations may be predefined by other objects in the system including agent module 2028 or object creation module 2024. For example, agent module 2028 such as the DSS agent™ offered by MicroStrategy, may be used to create and define reports with filters and template combinations, and to establish the alert criteria that are to be used for an alert service.

Next, in step 2120, the duration of the service is input by the user. Service duration indicates the starting and stopping dates for the service. The start date is the base line for the scheduled calculation, while the end date indicates when the broadcast will cease to be sent. The user has the option of starting the service immediately or waiting until some time in the future. Various calendaring features may be provided to enable the user to easily select these start and stop dates. For example, a calendar that specifies a date with pull-down menus that allow the users to select a month and year may be provided according to known methods of selecting dates in such programs as electronic calendaring programs and scheduling programs used in other software products. One specific aid that may be provided is to provide a calendar with a red circle indicating the present date and a blue ellipse around the current numerical date in each subsequent month to more easily allow the user to identify monthly intervals. Other methods may also be used.

Next, in step 2122, the user selects the schedule for the service. According to one embodiment, predefined schedules for services may be provided or the user may choose

to customize the schedule for the service. If the user desires to create a new schedule, a module may be opened to enable the user to name the schedule and to set the parameters for the schedule. Schedules may be run on a several-minute, hourly, daily, monthly, semi-annual or annual basis, all depending upon what frequency is desired.

The next step, step 2123, may be performed to enable the user to specify the content of a service. The content of a service is the various information reports and template/filter combinations that the server system 2014 processes using the data in data warehouse 2012 in order to provide the output requested for that particular service. The content of a service may comprise many different items or combination of items to suit the user's needs. For example, the user may be able to include a text grid, an agent alert, a web uniform resource location (URL), a spreadsheet container, a new sheet container, a text container, a text message, contents from a text file, or a file attachment. According to one embodiment, the system may organize these various contents into containers. A broadcast container may comprise the highest level container under which all content pieces reside. A grid may comprise an element that is associated with a report or a template/filter combination. The grid may be bound via a macro to a report/filter and template combination. An agent alert may be associated with a particular report that is therefore incorporated within the service. Any report available on agent module 2028 may be selected. The web URL item may be associated with the report through network output module 2022 that specified that URL for the particular report. A spreadsheet container may be the parent of an embedded spreadsheet attachment. When created, a

particular spreadsheet may be included as a child. Additionally, markup language (*e.g.*, XML and/or HTML) documents may also be included.

After the user has named the service, selected the type, duration, and schedule for the service, the user may select the personalization type in step 2124. For example, the user may select an option to either prevent personalization, require personalization, or allow personalize optionally. Upon completion of these steps, the service may be stored by service definition module 2042 in a database structure to enable users to retrieve predefined services to subscribe to these services through subscription interface module 2052.

Method 2210 may also comprise an error condition step. An error condition step may be used to enable users to specify "error" conditions and actions. For example, an "error" condition may be a user notification that a server is "down" or that there is no data to be returned. A user may specify particular actions to be performed by the system in response to one or more error conditions. For example, a user may specify a "server" error (*e.g.*, not responding) and indicate a particular action to be performed in response to a "server not responding" error (*e.g.*, reattempt in a predetermined time). Various other conditions and actions may be specified.

Broadcast module 2020 may also comprise a text-to-speech module that converts text into speech for output to speech devices such as telephones, answering machines, phone mail, and the like. Technologies for converting text to speech are known but the implementation of such a system in connection with a OLAP broadcast system has not been implemented until the present invention.

Further, a generated output message may comprise HTML Mail according to an embodiment of the present invention. A HTML Mail output module may be provided for generating output in HTML Mail format to enable viewers to receive more usable electronic mail messages. HTML Mail generation devices are known. The present invention provides general and customized HTML mail output by using, for example, the universal web data formatting capabilities of XML/XSL based design.

The present invention may convert reports into natural language outputs for more understandable information supply to subscribers. Pre-structured sentences and data fills and dynamic sentence construction may be provided as functionality in broadcast module 2020. The natural language capabilities of the agent module may be leveraged in this embodiment.

The system described may also comprise a portion of a larger decision support system 2010 as depicted in Fig. 12. System 2010 may comprise a data warehouse 2012, a server system 2014, an architect module 2016, an administrator module 2018, a broadcast module 2020, a network output module 2022, a plurality of user systems 2026, and an object creation module 2024. User systems 2026 may comprise an agent module 2028 as described above.

Agent module 2028 may enable a user access to the contents of data warehouse 2012 to provide detailed analysis on an *ad hoc* basis. One of the advantages of DSS Agent™ includes its use of a ROLAP architecture on server system 2014 and a RDBMS in data warehouse 2012 to provide a more scaleable environment. Through DSS Agent™, a user can "drill down." Drilling down allows the user to dynamically change

the level of detail in a report to a lower level attribute so that the resulting report displays data with a greater level of detail. For example, one can drill down from year to month to week to day. DSS Agent™ also enables users to "drill up" to a higher level attribute. Drilling up summarizes the selected data to a higher level total. For example, one can drill from day to week to month to year. DSS Agent™ also enables a user to "drill within." Drilling within allows a user to go to a different hierarchy within the same dimension. Drilling within is often used to examine the characteristics of selected data. For example, drilling within enables a user to drill from item to color when looking at a particular retail item such as an automobile, clothing or the like. Drilling across allows the user to drill to an altogether different dimension. For example, one can drill across from a region to a month. Accordingly, through use of agent module 2028, server system 2014, and data warehouse 2012, drilling is a powerful tool that is easily implemented using a ROLAP architecture which is not as easily accessible in MOLAP.

Architect module 2016 may comprise a module that enables developers to create and maintain data and metadata in data warehouse 2012. Metadata may be considered to be data about data, such as data element descriptions, data type descriptions, attributes/property descriptions, range/domain descriptions, and process/method descriptions. Data and metadata stored in data warehouse 2012 may thus be modified and organized by architect module 2016. According to one embodiment of the invention, architect module 2016 may comprise a software package known as DSS Architect™ offered by MicroStrategy.

Administrator module 2018 may comprise a module for facilitating the development, deployment, and management of data warehouse applications supporting large volumes of users over various distribution mechanisms. Administrator module 2018 may comprise an object manager and a warehouse monitor. The object manager allows objects to be shared across databases for easy migration from development to production. The warehouse monitor provides performance monitoring and management tools to support thousands of users across a distributive database environment. The warehouse monitor collects statistics for the purpose of identifying performance bottlenecks, warehouse tuning, cost analysis and various other purposes. According to one embodiment of the invention, administrator module 2018 may comprise a module known as DSS Administrator™ offered by MicroStrategy.

Server system 2014 may also connect to an object creation module 2024. Object creation module 2024 may comprise an open object linking and embedding ("OLE") application program interface ("API") for custom decision support development. According to one embodiment of the invention, object creation module 2024 may comprise a software module known as DSS Objects™ offered by MicroStrategy. Additionally, custom applications may interface with object creation module 2024 including Delphi, Visual Basic, and C++ programming modules.

User systems 2026 may also include a report writing module 2030, an executive module 2032, and a spreadsheet module 2034. Report writing module 2026 may comprise an OLAP report writer. Executive module 2032 may comprise a module design tool for developing custom EIS applications. This module is a design tool for developing

briefing books that provide high level users with a series of views that describe their business. Once created, end users can access briefing books through agent module 2028 in EIS mode. Such a system is easily implemented with agent module 2028 by compiling sets of analyses into dynamic pages that immediately focus users on their key business drivers. One embodiment of executive module 2032 may comprise software known as DSS Executive™ offered by MicroStrategy.

Spreadsheet module 2034 may comprise an add-on to existing spreadsheet programs or may comprise an entirely new spreadsheet program. Spreadsheet module 2034 may enable reports and analyses generated from agent module 2028 to be presented in a traditional spreadsheet program format to enable users to view results in preexisting front-end interfaces. Spreadsheet module 2034 may comprise the Microsoft Excel™ spreadsheet program offered by Microsoft and/or an Excel™ Add-In program offered by MicroStrategy.

Another module for accessing content of server system 2014 may comprise a network output module 2022. Network output module 2022 enables user system 2026 access to server system 2014 and data warehouse 2012 without requiring an additional agent module 2028 to be stored on user system 2026. Instead, user system 2026 may have a user interface module 2038 residing thereon. User interface module 2038 may comprise any module that enables a user system, such as user system 2026, to interface with network output module 2022 over a network 2036. According to one embodiment of the invention, network 2036 may comprise an intranet, the Internet or other developed Internet-type networks. Further, user interface module 2038 may comprise any standard

browser module such as Microsoft Internet Explorer™, Netscape Navigator™ or other. As many user systems 2026 already have a user interface module 2038 stored and operating thereon, network output module 2022 offers the advantage of enabling users access to server system 2014 and data warehouse 2012 without learning to operate a new module such as agent module 2028. One embodiment of network output module 2022 may comprise a web-based module called DSS Web™ offered by MicroStrategy. Accordingly, in one embodiment, a user can access server system 2014 through a standard web browser, such as Microsoft Internet Explorer™, or over the Internet through network output module 2022, such as DSS Web™.

In this embodiment, network output module 2022 may comprise a World Wide Web tool used in conjunction with server system 2014 for allowing users to deploy data warehouse/decision support applications over the Internet using industry standard World Wide Web browsers as a client. As a result, a user can access the data warehouse with little or no client maintenance, little or no software to install, and only a small amount of additional training while still maintaining all of the capabilities of agent module 2028. One embodiment of network output module 2022 comprises DSS Web™ offered by MicroStrategy. This embodiment provides a broad array of options for viewing information sets, such as spreadsheet grids and a wide variety of graphs. Through this module's reporting capabilities, users receive key elements of a report in easily interpretable, plain language messages. This module also allows users to "drill" dynamically to a lower level of detail to view the underlying information or to create and save new analyses. For sensitive information, this module provides security plug-ins that

allow the user to extend the standard security functionality with additional user authentication routines. This module may also provide an API that allows users to customize, integrate, and imbed this functionality into other applications. For example, a data syndicator for health care information may utilize this module with a customized interface to sell access to health care information to Health Maintenance Organizations, hospitals, pharmacies, etc.

An additional embodiment is provided for delivery of voice content. It is also recognized that this embodiment leverages the two-way devices to enable users to respond to information provided from the broadcast of information.

According to one embodiment of the present invention, a system is provided for automatic, interactive, real-time, voice transmission of OLAP output to one or more subscribers. For example, subscribers may be called by the system, and have content delivered audibly over the telephone or other voice-enabled terminal device. During the IVB, information may be exchanged between the system and a subscriber. The system conveys content to the subscriber and, the subscriber may respond by pressing one or more buttons on a telephone touch pad dial (or other input mechanism) to hear more information, to exercise options, or to provide other responses. This interaction shapes the structure of a basic exchange between the system and the subscriber. During or after the call is terminated, the subscriber's responses may be stored and processed (*e.g.*, by other applications).

According to one embodiment of the present invention, a method for automatic, interactive, real-time, voice transmission of OLAP output to one or more subscribers is

provided. Figure 1a depicts a flow chart of a method for automatic, interactive, real-time, voice transmission of OLAP output according to one embodiment. The method begins in step 110 with the creation of a voice service (*e.g.*, by a system administrator or user). A voice service is created using, for example, a voice service wizard which may comprise a series of interfaces. One embodiment of a method for creating a voice service is explained in more detail below in conjunction with Figure 1b. One embodiment of a voice service wizard is explained below in conjunction with Figure 3b.

After a voice service is created, users may subscribe or be subscribed to the voice service (step 120), for example, by using a subscription interface. According to one embodiment, users may subscribe to an existing voice service over the telephone or by web-based subscription. A user may also be subscribed programmatically. In other embodiments, a user may subscribe to a voice service via electronic mail. Not every voice service created in step 110 is available for subscription. More specifically, according to another embodiment, only a user with appropriate access, such as the creator of the service, is allowed to subscribe himself or others to a service. Such a security feature may be set when the voice service is created.

In step 130, a scheduling condition or other predetermined condition for the voice services is monitored to determine when they are to be executed. That is, when a voice service is created or subscribed to, the creator or user specifies when the voice service is to be executed. A user may schedule a voice service to execute according to the date, the time of day, the day of the week, etc. and thus, the scheduling condition will be a date, a time, or a day of the week, either one time or on a recurring basis. In the case of an alert

service, discussed in more detail below, the scheduling condition will depend on satisfaction of one or more conditions. According to one embodiment, the condition(s) to be satisfied is an additional scheduling condition. According to another embodiment, to another embodiment, a service may be executed "on command" either through an administrator or programmatically through an API. Scheduling of voice services is discussed in more detail below.

The method continues monitoring the scheduling condition for voice services until a scheduling condition is met. When a scheduling condition is met, that voice service is executed. The execution of a voice service involves, inter alia, generating the content for the voice service, and structuring the voice service to be telecast through a call server. The execution of a voice service is explained in detail in conjunction with Figure 1c.

An example of a telecast is as follows.

PERSONALIZED GREETING

Hello Joe, this is your stock update.

PIN VERIFICATION

Please enter your six digit PIN number

(Joe enters his PIN, using the keypad dial on his telephone.)

MENU OPTIONS

Your portfolio was up by $1000 today.

Please select:

1. To get a portfolio stock update

2. To conduct a transaction

(Joe presses 2)

SUB MENU

Thank you, Joe! Please select a ticker.

1. PQT

2. TQP

3. Listen to options again

4. Return to main menu

(Joe presses 1.)

SUB MENU

Would you like to buy or sell stock? Please press:

1. To sell stock

2. To buy stock

(Joe presses 1.)

SUB MENU

How many shares of PQT would you like to sell today? Please press:

1. To sell 50 shares

2. To sell 100 shares

3. To sell 200 shares

4. To sell another quantity

(Joe presses 2.)

SUB MENU

You selected 2. You want to sell 100 shares of PQT. Please press:

1. If this is correct

2. If this is incorrect

3. If you want to change the number of shares you want to buy.

(Joe presses 1.)

END VOICE SERVICE/TERMINATE TELECAST

Thank you for using our voice interactive broadcasting service, Joe. We will call you

back when your transaction is completed. Good-bye.

As can be seen from the above sample during an IVB, the user is presented with information, *e.g.*, the status of his portfolio, and is presented options related to that report, *e.g.*, the option to buy or sell stock.

According to one embodiment, a voice service is constructed using service wizard. A voice service is constructed using several basic building blocks, or elements, to organize the content and structure of a call. According to one embodiment, the building blocks of a voice service comprise elements of a markup language. According to one particular embodiment, elements of a novel markup language based on XML (TML) are used to construct voice services. Before explaining how a telecast is constructed, it will be helpful to define these elements.

The DIALOG element is used to define a unit of interaction between the user and the system and it typically contains one or more of the other elements. A DIALOG can not be contained in another element.

The SPEECH element is used to define text to be read to a user.

The INPUT element is used to define a section of a DIALOG that contains interactive elements, *i.e.*, those elements that relate to a response expected from a user

and its validation. An INPUT element may contain OPTION, PROMPT and ERROR elements.

An OPTION element identifies a predefined user selection that is associated with a particular input. According to one embodiment, OPTION elements are used to associate one or more choices available to a user with telephone keys.

A PROMPT element defines a particular input that is expected. According to one embodiment, a PROMPT element defines that a sequence or number of key presses from a telephone keypad is expected as input. Unlike an OPTION Element, a PROMPT Element is not associated with predefined user selections.

The PROMPT and OPTION elements may also be used to request user input using natural language. According to one embodiment, speech recognition technology is used to enable a user to respond to a PROMPT element or to select an OPTION element verbally by saying a number, e.g., "one.". The verbal response is recognized and used just as a keypress would be used. According to another embodiment, the user may provide a free form verbal input. For example, a PROMPT element may request that a user enter, e.g., the name of a business. In response the user speaks the name of a business. That spoken name is then resolved against predetermined standards to arrive at the input. Word spotting and slot filling may also be used in conjunction with such a PROMPT to determine the user input. For example, a PROMPT may request that the user speak a date and time, e.g., to choose an airline flight or to make a restaurant reservation. The user's spoken response may be resolved against known date and time formats to determine the input. According to another embodiment, a PROMPT is used to

request input using natural language. For instance, in conjunction with a voice service to be used to make travel plans, instead of having separate PROMPT elements request input for flight arrival, departure dates and locations, a single natural language PROMPT may ask, "Please state your travel plan." In response, the user states 'I'd like to go from Washington DC to New York city on the 3rd of January and return on the 3rd of February. This request would be processed using speech recognition and pattern matching technology to derive the user's input.

The ERROR element is used to define the behavior of the system if a user makes an invalid response such as touching a number that has not been associated with an OPTION element, or entering input that does not meet the criteria of a PROMPT element. A SYS-ERROR element defines a handler for certain events, such as expiration of the waiting time for a user response.

The FOR-EACH element is used to direct the system to loop through a list of variables *e.g.*, variables contained in a database report, or variables from a user input, to dynamically generate speech from data.

In addition to the elements described above, there are two features that maximize an administrator's ability to design voice services. Call Flow Reports enable an administrator to generate the structure of a call based on the content of an report *e.g.*, from an OLAP system or other data repository. For example, the options presented to a user in a PROMPT element may be made to correspond to the row of a data report. According to one embodiment, report data is converted into options by application of an

XSL (extensible style sheet language) style sheet. The result of this application is inserted into the static call structure when the voice service is executed.

The use of an XSL style sheet is a feature that maximizes an administrator's voice service building ability. As discussed above, they are used to create dynamic call structure that depends on data report output. They may also be used to generate a text string that comprises the message to be read to a user at any point in a call.

A method for creating a voice service according to one embodiment will now be explained in conjunction with Figure 2. The method begins in step 210 by naming the voice service. Then, in step 220 various scheduling parameters of the voice service are defined. In step 250 the service content is defined. And, in step 260, the personalization modes, or style properties are selected for the voice service.

According to one embodiment, in step 210, a voice service is named and a description of the voice service provided. By providing a name and description, a voice service may be uniquely identified. An interface is provided for prompting input of the name of the service to be created or edited. An input may also be provided for a written description. An open typing field would be one option for providing the description input. According to another embodiment, if an existing call service has been selected to edit, the service name field may not be present or may not allow modification.

In step 220, conditions for initiating the service are selected. This may include selecting and defining a service type. At least two types of services may be provided based on how the services are triggered. A first type of service is run according to a predetermined schedule and output is generated each time the service is run. A second

type of service, an alert service, is one that is run periodically as well, however, output is only generated when certain criteria is satisfied. Other service types may be possible as well. In one embodiment the administrator is prompted to choose between a scheduled service or an alert service. An interface may provide an appropriate prompt and some means for selecting between a scheduled service and an alert service. One option for providing the input might be an interface with a two element toggle list.

In one embodiment, a set of alert conditions is specified to allow the system to evaluate when the service should be initiated if an alert type service has been selected. In one embodiment, a report or a template/filter combination upon which the alert is based is specified. Reports and template/filter combinations may be predefined by other objects in the system including an agent module or object creation module. According to one embodiment, an agent module, such as DSS agent™ offered by MicroStrategy, may be used to create and define reports with filters and template combinations, and to establish the alert criteria for an alert service. According to another embodiment, an interface is be provided which includes a listing of any alert conditions presently selected for the voice service. According to this embodiment, the interface may comprise a display window. A browse feature may take the user to a special browsing interface configured to select a report or filter-template combination. One embodiment of an interface for selecting reports and filter-template combinations is described below. Once a report or filter and template combination is chosen, the alerts contained in the report or filter and template combination may be listed in the display window of the interface.

Sub Q3

In step 220, the schedule for the service is also selected. According to one embodiment, predefined schedules for voice services may be provided or a customized schedule for the voice service may be created. If a new schedule is to be created, a module may be opened to enable the schedule name and parameters to be set. Schedules may be run on a several-minute, hourly, daily, monthly, semi-annual, annual or other bases, depending upon what frequency is desired. According to one embodiment, an interface is provided that allows the administrator to browse through existing schedules and select an appropriate one. The interface may provide a browsing window for finding existing schedule files and a "new schedule" feature which initiates the schedule generating module. In one embodiment, schedules may not be set for alert type services. However, in some embodiments, a schedule for evaluating whether alert conditions have been met may be established in a similar manner.

Sub Q4

In step 220, the duration of the service is also set. Service duration indicates the starting and stopping dates for the service. Setting a service duration may be appropriate regardless of whether a scheduled service or alert type service has been selected. The start date is the base line for the scheduled calculation, while the end date indicates when the voice service will no longer be sent. The service may start immediately or at some later time. According to one embodiment, interface is provided to allow the administrator to input start and end dates. The interface may also allow the administrator to indicate that the service should start immediately or run indefinitely. Various calendar features may be provided to facilitate selection of start and stop dates. For example, a calendar that specifies a date with pull-down menus that allow selection of a day, month and year

may be provided according to known methods of selecting dates in such programs as electronic calendar programs and scheduling programs used in other software products. One specific aid that may be provided is to provide a calendar with a red circle indicating the present date and a blue ellipse around the current numerical date in each subsequent month to more easily allow the user to identify monthly intervals. Other methods may also be used.

In step 220, a voice service may also be designated as a mid-tier slicing service. In one embodiment, mid-tier slicing services generate content and a dynamic subscriber list in a single query to an OLAP system. According to one embodiment, in a mid-tier slicing service a single database query is performed for all subscribers to the service. The result set developed by that query is organized in a table that contains a column that indicates one or more users that each row of data is applicable to.

In step 250, the content of the voice service is defined. Defining the content of the voice service may include selecting the speech to be delivered during the voice service broadcast (content), the structure of dialogs, menus, inputs, and the background procedures which generate both content and structure. In one embodiment, defining voice service content establishes the procedures performed by the vss server to assemble one or more active voice pages in response to initiation of the voice service. According to one embodiment, defining service content involves establishing a hierarchical structure of TML elements which define the structure and content of a voice service. All of the elements in a given service may be contained within a container.

The personalization type is selected in step 260. Personalization type defines the options that the administrator will have in applying personalization filters to a voice service. According to one embodiment, a personalization filter is a set of style properties that can be used to determine what content generated by the service will be delivered to the individual user and in what format it will be delivered. In one embodiment, personalizing the delivery format may include selection of style properties that determine the sex of the voice, the speed of the voice, the number of call back attempts, etc. Personalization filters may exist for individual users, groups of users, or types of users. According to one embodiment, personalization filters may be created independent of the voice service. According to this embodiment, a voice service specifies what filters are used when generating IVBs. Some personalization type options may include: allowing no personalization filters; allowing personalization filters for some users, but not requiring them; and requiring personalization filters for all interactive voice broadcasts made using the service.

According to one embodiment, specifying personalization type is accomplished by administrator input through an interface. The interface may offer a toggle list with the three options: required personalization, optional personalization, and no personalization.

The voice service may be stored in a database structure to enable users to retrieve predefined voice services and to subscribe to these services, for example, through subscription interfaces explained in conjunction Figures 3a-3c or otherwise. An interface informing the administrator that creation of the voice service is complete may also be provided.

According to one embodiment, the method of Figure 1b also comprises an error condition step. An error condition step may be used to enable administrators to specify "error" conditions and the handling of those conditions. For example, an "error" condition may comprise a notification that a server is "down" or that there is no data to be returned. An administrator may specify particular actions to be performed by the system in response to one or more error conditions. For example, an administrator may specify an "addressing" error (*e.g.*, disconnected number) and indicate a particular action to be performed in response to an "addressing" error (*e.g.*, notify system administrator). Other error conditions might include: an alert report encountering an error and returning no data; a subscriber lacking the required personalization filter for the service; errors occurring in the generation of one or more reports; or reports returning no data. Various other conditions and actions may be specified. Certain error conditions may be predetermined for the system, but an administrator may have reasons for supplementing or diverging from the predetermined error conditions. According to one particular embodiment, error conditions are specified using the ERROR and SYS-ERROR elements.

In one embodiment, setting error conditions may be accomplished using an error handling interface. The interface may allow the administrator to select either default error handling, or to customize error handling using a module for defining error handling. If default handling is selected, the system uses established settings. If customized handling is chosen, the user may use a feature to access the appropriate interface for the error handling module.

Sub Q5

Servers may have limited capacity to perform all of the actions required of them simultaneously, the method of Figure 1b comprises a step for prioritizing the execution and delivery of voice services. Prioritization may establish the order in which the voice service system allocates resources for processing voice service and delivering the IVB. According to one embodiment, assigning priority to a voice service establishes priority for queries to the database system, formatting the voice service, or IVBs. Any criteria may be used for establishing priority . According to one embodiment, priority is established based on service content. According to another embodiment, priority is based on service destination. According to another embodiment, priority may be established based on the type of voice service, *i.e.*, alert vs. scheduled. Any number of procedures or criteria for denoting relative importance of service delivery may be established.

In one embodiment, an interface is provided for defining the priority of the voice service being created or edited. According to one embodiment, the interface comprises a screen including option boxes with pull down menus listing the number of different prioritization options.

Another aspect of the invention relates to a method for executing a voice service. Figure 1c depicts one example of a flow chart for executing a voice service. In step 310, the content of a voice service is generated. In step 320, the call structure of a telecast is created via Active Voice Pages. In step 330, the AVPs are put in a call database for processing *e.g.*, in a call queue. In step 340, the call request is processed and an interactive voice broadcast with the user is implemented. In step 350, user's responses

are written back to the voice service system (*e.g.*, the Active Voice Page). Each of these steps will be explained in more detail below.

According to one embodiment, content is created in step 310 as follows. A voice service execution begins by running scheduled reports, queries or by taking other action to determine whether the service should be sent. The subscribers for the service are then resolved. Datasets are generated for each group of subscribers that has unique personalization criteria.

Call structure may be created (step 320) as follows. An AVP contains data at various hierarchical content levels (nodes) that can be either static text or dynamic content. Static text can be generated *e.g.*, by typing or by incorporating a text file. Dynamic content may be generated *e.g.*, by inserting data from a data report using a grid an/or an XSL stylesheet. Moreover, content is not limited to text based information. Other media, such as, sound files, may be incorporated into the AVP. The call data (for example, at a particular level) may be the text that is converted to speech and played when the recipient encounters the node.

According to another embodiment, call content may include "standard" active voice pages that are generated and inserted into a database or Web Server where the pages are periodically refreshed. According to one particular embodiment, the active voice page that is generated for a user contains links to these standard active voice pages. The links may be followed using a process similar to web page links.

The call structure may comprise either a static structure that is defined in the voice service interfaces *e.g.*, by typing text into a text box and/or a dynamic structure generated

by grid/XSL combinations. The dynamic structure is merged with static structure during the service execution. A single call structure is created for each group of users that have identical personalization properties across all projects because such a group will receive the same content.

After a call structure is generated, in step 330, it is sent to a call database *e.g.*, call database 1811 shown in Figure 3calong with the addresses and style properties of the users. The style properties govern the behavior of a call server 18 in various aspects of the dialog with a user. Call server 18 queries call database 1811 for current call requests and places new call requests in its queue.

In step 340, a call request is processed. A call is implemented on call server 18 using one of several ports that are configured to handle telephone communication. When a port becomes available, the call request is removed from the queue and the call is made to the user. As the user navigates through an active voice page, *e.g.*, by entering input using the key pad or by speaking responses, call/content is presented by converting text to speech in text-to-speech engine 1814. User input during the call may be stored for processing. According to another embodiment, user responses and other input may also be used to follow links to other active voice pages. For example, as explained above, "standard" active voice pages may be generated and inserted into a database or Web Server. Then, when a user's voice service is delivered, that voice service may contain links to information that may be accessed by a user. A user may access those standard active voice pages by entering input in response to OPTION or PROMPT elements.

In step 350, user responses are stored by the system. According to one embodiment, user responses are stored in a response collection defined by the active voice page. A voice service may specify that a subscriber return information during an IVB so that another application may process the data. For instance, a user may be prompted to purchase a commodity and be asked to enter or speak the number of units for the transaction. During or after an IVB, the subscriber's responses are written to a location from which they can be retrieved for processing (*e.g.*, by an external application).

Fig. 2 is an example of an IVB with interactive call flow. An IVB usually contains a greeting message that addresses the targeted user, identifies the name of the calling application, and states the purpose of the call and/or presents summary metrics. The voice service system can also implement a PIN verification protocol, if this layer of security is required. The main menu structure of an IVB can contain a number of options that lead to sub-menu structures. A menu can also contain prompts for the user to enter numerical information using a telephone touch pad dial. A node along the hierarchical menu structure may have options to return the user to a higher level.

Fig. 3 depicts an embodiment of a system according to one embodiment of the present invention. Preferably, the system comprises database system 12, a DSS server 14, voice service server 16, a call server 18, subscription interface 20, and other input/files 24.

Database system 12 and DSS server 14 comprise an OLAP system that generates user-specified reports from data maintained by database system 12. Database system 12 may comprise any data warehouse or data mart as is known in the art, including a

relational database management system ("RDBMS"), a multidimensional database management system ("MDDBMS") or a hybrid system. DSS server 14 may comprise an OLAP server system for accessing and managing data stored in database system 12. DSS server 14 may comprise a ROLAP engine, MOLAP engine or a HOLAP engine according to different embodiments. Specifically, DSS server 14 may comprise a multithreaded server for performing analysis directly against database system 12. According to one embodiment, DSS server 14 comprises a ROLAP engine known as DSS Server™ offered by MicroStrategy.

Voice service server (VSS) 16, call server 18 and subscription interface 20 comprise a system through which subscribers request data and reports *e.g.*, OLAP reports through a variety of ways and are verbally provided with their results through an IVB. During an IVB, subscribers receive their requested information and may make follow-up requests and receive responses in real-time as described above. Although the system is shown, and will be explained, as being comprised of separate components and modules, it should be understood that the components and modules may be combined or further separated. Various functions and features may be combined or separated

Subscription interface 20 enables users or administrators of the system to monitor and update subscriptions to various services provided through VSS 16. Subscription interface 20 includes a world wide web (WWW) interface 201, a telephone interface 202, other interfaces as desired and a subscriber API 203. WWW interface 201 and telephone interface 202 enable system 100 to be accessed, for example, to subscribe to voice services or to modify existing voice services. Other interfaces may be used. Subscriber

API 203 provides communication between subscription interface 20 and VSS 16 so that information entered through subscription interface 20 is passed through to VSS 16.

Subscription interface 20 is also used to create a subscriber list by adding one or more subscribers to a service. Users or system administrators having access to VSS 16 may add multiple types of subscribers to a service such as a subscriber from either a static recipient list (SRL) (*e.g.*, addresses and groups) or a dynamic recipient list (DRL) (described in further detail below). The subscribers may be identified, for example, individually, in groups, or as dynamic subscribers in a DRL. Subscription interface 20 permits a user to specify particular criteria (*e.g.*, filters, metrics, etc.) by accessing database system 12 and providing the user with a list of available filters, metrics, etc. The user may then select the criteria desired to be used for the service. Metadata may be used to increase the efficiency of the system.

A SRL is a list of manually entered names of subscribers of a particular service. The list may be entered using subscription interface 20 or administrator console 161. SRL entries may be personalized such that for any service, a personalization filter (other than a default filter) may be specified. A SRL enables different personalizations to apply for a login alias as well. For example, a login alias may be created using personalization engine 1632. Personalization engine 1632 enables subscribers to set preferred formats, arrangements, etc. for receiving content. The login alias may be used to determine a subscriber's preferences and generate service content according to the subscriber's preferences when generating service content for a particular subscriber.

A DRL may be a report which returns lists of valid user names based on predetermined criteria that are applied to the contents of a database such as database system 12. Providing a DRL as a report enables the DRL to incorporate any filtering criteria desired, thereby allowing a list of subscribers to be derived by an application of a filter to the data in database system 12. In this manner, subscribers of a service may be altered simply by changing the filter criteria so that different user names are returned for the DRL. Similarly, subscription lists may be changed by manipulating the filter without requiring interaction with administrator console 161. Additionally, categorization of each subscriber may be performed in numerous ways. For example, subscribers may be grouped via agent filters. In one specific embodiment, a DRL is created using DSS Agent™ offered by MicroStrategy.

VSS 16 is shown in more detail in Figure 3b. According to one embodiment, VSS 16 comprises administrator console 161, voice service API 162 and backend server 163. Administrator console 161 is the main interface of system 100 and is used to view and organize objects used for voice broadcasting. Administrator console 161 provides access to a hierarchy of additional interfaces through which a system administrator can utilize and maintain system 100. Administrator console 161 comprises system administrator module 1611, scheduling module 1612, exceptions module 1613, call settings module 1614, address handling module 1615, and service wizard 1616.

System administrator module 1611 comprises a number of interfaces that enable selection and control of the parameters of system 100. For example, system administrator module 1611 enables an administrator to specify and/or modify an email system,

supporting servers and a repository server with which system 100 is to be used. System administrator 1611 also enables overall control of system 100. For example, system administrator module is also used to control the installation process and to start, stop or idle system 100. According to one embodiment, system administrator 1611 comprises one or more graphical user interfaces (GUIs).

Scheduling module 1612 comprises a number of interfaces that enable scheduling of voice services. Voice services may be scheduled according to any suitable methodology, such as according to scheduled times or when a predetermined condition is met. For example, the predetermined condition may be a scheduled event (time-based) including, day, date and/or time, or if certain conditions are met. In any event, when a predetermined condition is met for a given service, system 100 automatically initiates a call to the subscribers of that service. According to one embodiment, scheduling module 1612 comprises one or more GUIs.

Exceptions module 1613 comprises one or more interfaces that enable the system administrator to define one or more exceptions, triggers or other conditions. According to one embodiment, exceptions module 1613 comprises one or more GUIs.

Call settings module 1614 comprises one or more interfaces that enable the system administrator to select a set of style properties for a particular user or group of users. Each particular user may have different options for delivery of voice services depending on the hardware over which their voice services are to be delivered and depending on their own preferences. As an example of how the delivery of voice services depends on a user's hardware, the system may deliver voice services differently

depending on whether the user 's terminal device has voice mail or not. As an example of how the delivery of voice services depends on a user's preferences, a user may chose to have the pitch of the voice, the speed of the voice or the sex of the voice varied depending on their personal preferences. According to one embodiment, call settings module 1614 comprises one or more GUIs.

Address handling module 1615 comprises one or more interface that enable a system administrator to control the address (*e.g.*, the telephone number) where voice services content is to be delivered. The may be set by the system administrator using address handling module 1615. According to one embodiment, address handling module 1615 comprises one or more GUIs.

Voice service wizard module 1616 comprises a collection of interfaces that enable a system administrator to create and/or modify voice services. According to one embodiment, service wizard module 1616 comprises a collection of interfaces that enable a system administrator to define a series of dialogs that contain messages and inputs and determine the call flow between these dialogs based on selections made by the user. The arrangement of the messages and prompts and the flow between them comprises the structure of a voice service. The substance of the messages and prompts is the content of a voice service. The structure and content are defined using service wizard module 1616.

Voice service API 162 (*e.g.*, MicroStrategy Telecaster Server API) provides communication between administrator console 161 and backend server 163. Voice Service API 162 thus enables information entered through administrator console 161 to be accessed by backend server 163 (*e.g.*, MicroStrategy Telecaster Server).

Backend server 163 utilizes the information input through administrator console 161 to initiate and construct voice services for delivery to a user. Backend server 163 comprises report formatter 1631, personalization engine 1632, scheduler 1633 and SQL engine 1634. According to one embodiment, backend server 163 comprises MicroStrategy Broadcast Server. Report formatter 1631, personalization engine 1632, and scheduler 1633 operate together, utilizing the parameters entered through administrator console 161, to initiate and assemble voice services for transmission through call server 18. Specifically, scheduler 1633 monitors the voice service schedules and initiates voice services at the appropriate time. Personalization engine 1632 and report formatter 1631 use information entered through service wizard 1616, exceptions module 1613, call settings module 1614, and address module 1615, and output provided by DSS server 14 to assemble and address personalized reports that can be sent to call server 18 for transmission. According to one embodiment, report formatter 1631 includes an XML based markup language engine to assemble the voice services. In a particular embodiment, report formatter includes a Telecaster Markup Language engine offered by MicroStrategy Inc. to assemble the call content and structure for call server 18.

SQL engine 1634 is used to make queries against a database when generating reports. More specifically, SQL engine 1634 converts requests for information into SQL statements to query a database.

Repository 164 may be a group of relational tables stored in a database. Repository 164 stores objects which are needed by system 100 to function correctly.

More than one repository can exist, but preferably the system 100 is connected to only one repository at a time.

According to one embodiment, a call server 18 is used to accomplish transmission of the voice services over standard telephone lines. Call server 18 is shown in more detail in Figure 3c. According to one embodiment, call server 18 comprises software components 181 and hardware components 182. Software components 181 comprise call database 1811, mark-up language parsing engine 1812, call builder 1813, text-to-speech engine 1814, response storage device 1815 and statistic accumulator 1816.

Call database 1811 comprises storage for voice services that have been assembled in VSS 16 and are awaiting transmission by call server 18. These voice services may include those awaiting an initial attempt at transmission and those that were unsuccessfully transmitted (*e.g.*, because of a busy signal) and are awaiting re-transmission. According to one embodiment, call database 1811 comprises any type of relational database having the size sufficient to store an outgoing voice service queue depending on the application. Call database 1811 also comprises storage space for a log of calls that have been completed.

Voice services stored in call database 1811 are preferably stored in a mark-up language. Mark-up language parsing engine 1812 accepts these stored voice services and separates the voice services into parts. That is, the mark-up language version of these voice services comprises call content elements, call structure elements and mark-up language instructions. Mark-up language parsing engine 1812 extracts the content and structure from the mark-up language and passes them to call builder 1813.

Call builder 1813 is the module that initiates and conducts the telephone call to a user. More specifically, call builder dials and establishes a connection with a user and passes user input through to markup language parsing engine 1812. In one embodiment, call builder 1813 comprises "Call Builder" software available from Call Technologies Inc. Call builder 1813 may be used for device detection, line monitoring for user input, call session management, potentially transfer of call to another line, termination of a call, and other functions.

Text-to-speech engine 1814 works in conjunction with mark-up language parsing engine 1812 and call builder 1813 to provide verbal communication with a user. Specifically, after call builder 1813 establishes a connection with a user, text-to-speech engine 1814 dynamically converts the content from mark-up language parsing engine 1812 to speech in real time.

A voice recognition module may be used to provide voice recognition functionality for call server 181. Voice recognition functionality may be used to identify the user at the beginning of a call to help ensure that voice services are not presented to an unauthorized user or to identify if a human or machine answers the call. This module may be a part of call builder 1813. This module may also be used to recognize spoken input (say "one" instead of press "1"), enhanced command execution (user could say "transfer money from my checking to savings"), enhanced filtering (instead of typing stock symbols, a user would say "MSTR"), enhanced prompting, (saying numeral values).

User response module 1815 comprises a module that stores user responses and passes them back to intelligence server 16. Preferably, this is done within an AVP.

During a telephone call, a user may be prompted to make choices in response to prompts by the system. Depending on the nature of the call, these responses may comprise, for example, instructions to buy or sell stock, to replenish inventory, or to buy or rebook an airline flight. User response module 1815 comprises a database to store these responses along with an identification of the call in which they were given. The identification of the call in which they were given is important to determining what should be done with these responses after the call is terminated. User responses may be passed back to intelligence server 16 after the call is complete. The responses may be processed during or after the call, by the system or by being passed to another application.

Statistics accumulator 1816 comprises a module that accumulates statistics regarding calls placed by call builder 1813. These statistics including, for example, the number of times a particular call has been attempted, the number of times a particular call has resulted in voice mail, the number of times a user responds to a call and other statistics, can be used to modify future call attempts to a particular user or the structure of a voice service provided to a particular user. For example, according to one embodiment, statistics accumulator 1816 accumulates the number of times a call has been unsuccessfully attempted by call builder 1813. This type of information is then used by call server 18 to determine whether or not the call should be attempted again, and whether or not a voice mail should be left.

Call server 18 also comprises certain hardware components 182. As shown in Figure 1c, hardware components 182 comprise processor 1821 and computer telephone module 1822. According to one embodiment, processor 1821 comprises a Pentium II

processor, available from Intel, Inc. Module 1822 provides voice synthesis functionality that is used in conjunction with Text to Speech engine 1814 to communicate the content of voice services to a user. Module 1822 preferably comprises voice boards available from Dialogic, Inc. Other processors and voice synthesizers meeting system requirements may be used.

Q8 contd.

The system and method of the present invention may form an integral part of an overall commercial transaction processing system.

Sub Q9

According to one embodiment of the present invention, a system and method that enable closed-loop transaction processing are provided. The method begins with the deployment of an IVB by executing a service. As detailed above, this includes generating the content and combining this with personalization information to create an active voice page. Call server 18 places a call to the user. During the call, information is delivered to the user through a voice-enabled terminal device (e.g., a telephone or cellular phone).

During the IVB, a user may request a transaction, service, further information from the database or other request, *e.g.*, based on options presented to the user. These will generically be referred to as transactions. The request may be, but is not necessarily, based on or related to information that was delivered to the user. According to one embodiment, the request comprises a user response to a set of options and/or input of information through a telephone keypad, voice input or other input mechanism. According to another embodiment, the request can be made by a user by speaking the request. Other types of requests are possible.

According to one embodiment, the user responses are written to a response collection, which along with information stored in the active voice page, can be used to cause a selected transaction to be executed. According to one embodiment, the active voice page comprises an XML-based document that includes embedded, generic requests, *e.g.*, a request for a transaction, or a request for additional information (a database query). These embedded requests are linked with, for example option statements or prompts so that when a user enters information, the information is entered into the generic request and thus completes a specific transaction request. For example, in the example if a user exercises an option to buy a particular stock, that stock's ticker symbol is used to complete a generic "stock buy" that was embedded in the active voice page.

According to one embodiment, tokens are used to manage user inputs during the IVB. A token is a temporary variable that can hold different values during an IVB. When a user enters input, it is stored as a token. The token value is used to complete a transaction request as described above. According to one embodiment, the system maintains a running list of tokens, or a response collection, during an IVB.

In order to complete the requested transaction, the user responses (and other information from the active voice page) may need to be converted to a particular format. The format will depend, for example, on the nature and type of transaction requested and the system or application that will execute the transaction. For example, a request to purchase goods through a web-site may require the information to be in HTML/HTTP format. A request for additional information may require and SQL statement. A telephone-based transaction may require another format.

Therefore, the transaction request is formatted. According to one embodiment, the transaction is formatted to be made against a web-based transaction system. According to another embodiment, the transaction request is formatted to be made against a database. According to another embodiment, the transaction is formatted to be made against a telephone-based transaction system. According to another embodiment, the transaction is formatted to be made via e-mail or EDI. Other embodiments are possible.

In one embodiment, the formatted transaction request comprises an embedded transaction request. The system described in connection with Figures 1-3 provides interactive voice services using TML, a markup language based on XML. Using TML active voice pages are constructed that contain the structure and content for a interactive voice broadcast including, inter alia, presenting the user with options and prompting the user for information. Moreover in connection with OPTION and PROMPT elements, active voice pages also can include embedded statements such as transaction requests. Therefore, the formatting for the transaction request can be accomplished ahead of time based on the particular types of transactions the user may select.

For example, in connection with an exemplary stock purchase, an active voice page can include an embedded transaction request to sell stock in the format necessary for a particular preferred brokerage. The embedded statement would include predefined variables for the name of the stock, the number of shares, the type of order (market or limit, etc.), and other variables. When the user chooses to exercise the option to buy or sell stock, the predefined variables are replaced with information entered by the user in

response to OPTION or PROMPT elements. Thus, a properly formatted transaction request is completed.

In the system of Figures 1-3, TML parsing engine in call server 18 includes the functionality necessary to generate the properly formatted transaction request as described above. For example, in connection with the embodiment described above, the TML parsing engine shown in Figure 3c reads the active voice pages. When the TML parsing engine reads an OPTION element that includes and embedded transaction request, it stores the transaction request, and defines the necessary variables and variable locations. When the user exercises that OPTION, the user's input is received by the TML parsing engine and placed at the memory locations to complete the transaction request This technique could be used, for example, to generate a formatted transaction request for web-site.

According to another embodiment, where the transaction request is made via a natural language, voice request, a formatted transaction request can be generated in a number of ways. According to one embodiment, speech recognition technology is used to translate the user's request into text and parse out the response information. The text is then used to complete an embedded transaction request as described above. According to another embodiment, speech recognition software is used to translate the request to text. The text is then converted to a formatted request based on a set of known preferences.

A connection is established with the transaction processing system. This can be accomplished during, or after the IVB. According to one embodiment, the transaction processing system comprises a remotely located telephone-based transaction site. For

example, in the system shown in Figures 1-3, call server 18, through the TML parsing engine 1812, establishes a connection with a telephone-based transaction processing site.

According to another embodiment, the transaction processing system comprises a remotely based web-site. According to this embodiment, the formatted request includes a URL to locate the web-site and the system accesses the site through a web connection using the formatted request. Alternatively, the formatted request includes an e-mail address and the system uses any known email program to generate an e-mail request for the transaction.

After the connection is established, the transaction is processed by the transaction processing site and the user is notified of the status of the transaction. If the transaction is completed in real-time, the user may be immediately notified. If the transaction is executed after the IVB, the user may be called again by the system, sent an e-mail, or otherwise notified when the transaction has been completed.

According to one particular embodiment, the system comprises the interactive voice broadcasting system shown and described in Figures 1-3 and the transaction is accomplished in real-time. In this embodiment, confirmation of the transaction is returned to TML parsing engine 1812 shown in Figure 3 and translated to speech in text-to-speech engine 1814 and presented to the user during the IVB. More specifically, and similar to the process described with respect to embedded formatted transaction requests, TML also enables embedding of a response statement. Thus, when the transaction is processed and confirmation of the transaction is returned to the system, an embedded

confirmation statement is conveyed to the user through TML parsing engine 1812 after being converted to speech in text-to-speech engine 1814.

Figure 4 schematically depicts one example of how the system and method of the present invention would fit into such a commercial transaction processing system. Working from left to right in Figure 4, the system begins and ends with information stored in relational databases. One of the primary purposes of information is in making decisions. Thus, the information in the databases is most useful if provided to someone who desires it in a timely fashion.

A voice service system is provided to enable access to the information in the databases. The voice service system utilizes personalization information and personalized menus to construct AVPs pages that enable the information to be delivered to a user verbally. Moreover, the AVPs pages, not only enable information to be presented to the user. But, they also enable the user to provide information back to the voice service system for additional processing.

According to the embodiment shown in Figure 4, once the AVPs are constructed by voice service system, they are processed and the content is delivered to a user verbally in an IVB. Thus, call processing and text-to-speech technology are used to establish a telephone connection with a user and convert the active voice pages to speech for presentation to the user. As shown in Figure 4, the IVB may be delivered to a user in many devices, including a telephone, a mobile phone, voice mail, an answering machine or any other voice-enabled device.

During the IVB, depending on the content that is being delivered, control may be passed to an e-commerce application for the user to complete a transaction based on the information presented. For example, if the user has requested information about sales on a particular brand of merchandise, the user may be connected with a particular retailer in order to complete a transaction to buy a particular good or service. Information about this transaction is then added to the databases and thus may be advantageously accessed by other users.

It may not be economical for some potential users of a voice broadcasting system to buy and/or maintain their own telephony hardware and software as embodied in call server 18. In such a case, a voice service bureau may be maintained at a remote location to service users voice service requests. A voice service bureau and a method of using a voice service bureau according to various embodiments of the present invention is described in conjunction with Figures 5-6.

In one embodiment, a voice service bureau may comprise one or more call servers and call databases that are centrally located and enable other voice service systems to generate a call request and pass the call request to the VSB to execute a call. In this way the other voice service systems do not need to invest in acquiring and maintaining call data bases, call servers, additional telephone lines and other equipment or software. Moreover, the VSB facilitates weeding out usage of illegal numbers and spamming by number checking implemented through its web server.

A voice service bureau and a method of using a voice service bureau according to one embodiment are described in conjunction with Figures 5-6. Figure 5 depicts a

method of utilizing a voice service bureau according to one embodiment of the present invention. The method begins in step 810 with a request to place one or more telephone calls received through a computer network.

According to one embodiment, the voice service bureau is maintained at a location distant from the voice service system. Therefore, in order for a voice service to be processed by the voice service bureau, in step 810 the voice service is sent to the voice services bureau, preferably over some secure line of communication. According to one embodiment, the request is sent to the voice service bureau through the Internet using secure HTTPS. HTTPS provides a secure exchange of data between clients and the voice service bureau using asymmetric encryption keys based on secure server certificates. In another embodiment, SSL HTTP protocol is used to send a call request to the voice service bureau. Both of these protocols help ensure that a secure channel of communication is maintained between the voice service system and the voice service bureau. Other security techniques may be used.

When a request for a call or telecast is received, by the VSB, the request is authenticated by the voice service bureau in step 820. According to one embodiment, the authenticity of the request is determined in at least two ways. First, it is determined whether or not the request was submitted from a server having a valid, active server certificate. More specifically, requests may be typically received via a stream of HTTPS data. Each such request originating from a server with a valid server certificate will include an embedded code (i.e., server certificate) that indicates the request is authentic. In addition to the use of server certificates, each request may also be authenticated using

an identification number and password. Therefore, if the request submitted does not include a valid server certificate and does not identify a valid I.D./password combination, the request will not be processed. The step of authenticating also comprises performing any necessary decryption. According to one embodiment, any errors that are encountered in the process of decrypting or authenticating the call request are logged an error system and may be sent back to the administrator of the sending system. Other methods of authenticating the request are possible.

Each properly authenticated request is sent to a call server (step 830) and processed (step 840). According to one embodiment, the voice service bureau comprises a number of call servers. According to one embodiment, the calls are sent to a call database, and processed as set forth herein in conjunction with the explanation of call server 18.

One embodiment of a voice service bureau will now be explained in conjunction with Figures 6a-6c. Figure 6a depicts a system comprising a plurality of client side installations 91, a primary voice bureau 92, a system administrator 93, a backup voice service bureau 94, and a plurality of users 95. Client side installations 91 communicate with voice service bureau 92 through a computer network. Voice service bureau 92 communicates with users 95 through a voice network. According to one embodiment, the computer network comprises the internet and client side installations 91 communicate with voice service bureau 92 using HTTPS as described above, and the voice network comprises a public telephone network.

According to one embodiment, client side installations 91 are substantially identical to the system shown in Figure 4 except for the elimination of call server 18. In the system of Fig. 6a, the functionality of call server 18 is performed by VSB 92. As shown in this embodiment, VSB 92 can service multiple client side installations 91_1 to 91n. According to another embodiment, client-side installation functionality may be included within VSB 92. According to this embodiment VSB 92 constitutes a fully functional voice service that is accessible through email, telephone or other interfaces.

According to this embodiment, when voice services have been assembled by intelligence server 16, a request to have the voice services transmitted is sent via a secure network connection through the computer network shown to primary voice bureau 92 and backup voice service bureau 94 as described above. According to one embodiment, the request comprises a mark-up language string that contains the voice service structure and content and personal style properties and other information. As described above, voice bureau 92 authenticates the request, queues the voice services and sends telecasts to users 95 through the voice network.

Sub Q10

A block diagram of one embodiment of primary voice bureau 92 is shown in Figure 6b. According to this embodiment, primary voice bureau comprises routers 921, dual-homed servers 922, database servers 923, call database 924, backup storage 925, call servers 926, internal switch 927, and system administrator 928. Routers 921 receive call requests via a computer network and pass them along to one of the two dual-homed servers 922. Router 921 monitors activity on servers 922 and forwards call requests to one of the two depending on availability.

Dual-homed servers 922 comprise servers configured to receive and send HTTPS email. As part of their receiving function, dual-homed servers 922 are configured to perform the authentication processing described above. According to one embodiment, dual-homed servers 922 determine whether the incoming request originated from a server with an active server certificate and also determine if the request contains a valid I.D./password combination. Once dual-homed servers 922 have authenticated the incoming request, they forward the request to be queued in call database 924. As part of their sending function, dual-homed servers 922 are configured to format and send HTTPS email. As discussed above, during a telecast a user may request that further information be accessed from a database or that some transaction be performed. According to one embodiment, these user requests are forwarded back to the originating system via HTTPS email by dual-homed servers 922. Dual-homed servers 922 are load balanced to facilitate optimal performance and handling of incoming call requests.

Database servers 923, call database 924, and backup storage 925 together comprise a call request queuing system. Primary voice bureau 92 is configured to handle a large number of call requests. It may not be possible to process call requests as they arrive. Therefore, call requests are queued in call database 924. According to one embodiment, call database 924 comprises a relational database that maintains a queue of all call requests that need to be processed as well as logs of calls that have been processed. According to another embodiment, primary VSB 92 may include a failover measure that enables another system server to become the call database if call database 924 should fail.

Database servers 923 are configured to control access to call database 924. According to one embodiment, database servers may be optimized to generate SQL statements to access entries in call database at high speed. Database servers 923 also control storage of call requests and call logs in call database 924.

Call servers 926 each are configured to format and send telecasts. According to one embodiment, each of call servers 926 is substantially identical to call server 18 shown in Figure 3c. More specifically, each of call servers 926 receives requests for telecasts, parses the call content from the mark-language, establishes a connection with the user through phone lines 929, and receives user responses. According to one embodiment, call servers 926 comprise a clustered architecture that facilitates message recovery in the event of server failure.

Primary voice bureau 92 is controlled by system administrator 93 and internal switch 927. System administrator controls switch 927 and thus controls the flow of call requests to call database 924 from dual homed servers 922 and to call servers 926 from call database 924.

System administrator 93 is also configured to perform a number of other services for primary voice bureau 92. According to one embodiment, system administrator 93 also comprises a billing module, a statistics module, a service module and a security module. The billing modules tabulates the number of voice service requests that come from a particular user and considers the billing plan that the customer uses so that the user may be appropriately billed for the use of voice bureau 92. The statistics module determines and maintains statistics about the number of call requests that are processed

by voice bureau 92 and statistics regarding call completion such as, e.g., success, failed due to busy signal and failed due to invalid number. These statistics may be used, for example, to evaluate hardware requirements and modify pricing schemes. The security module monitors activity on voice bureau 92 to determine whether or not any unauthorized user has accessed or attempted to access the system. The service module provides an interface through which primary voice bureau 92 may be monitored, for example, to determine the status of call requests. Other service modules are possible. Moreover, although these services are described as distinct modules, their functionality could be combined and provided in a single module.

Backup voice service bureau 94 receives a redundant request for voice services. Backup voice service bureau 94 processes the requests only when primary voice service bureau is offline or busy. One embodiment of backup voice service bureau 94 is shown in Figure 6c. Backup voice bureau 94 comprises routers 941, HTTP server 942, database server 943, call server 946 and routers 947. Each of these components performs a function identical to the corresponding element in primary voice bureau 92. Router 947 replaces switch 927. Router 947 controls the forwarding of call requests to database server 943 for queuing in an internal database, and the forwarding of call requests to call server 946 from database server 943.

The systems and methods discussed above are directed to outbound broadcasting of voice services. Nevertheless, in certain situations, for example when the out bound telecast is missed, it is desirable to for a voice service system to enable inbound calling.

According to another embodiment, a method and system for providing integrated inbound and outbound voice services is disclosed.

A method for providing inbound access to voice services according to one embodiment of the present invention is shown in Figure 7. According to Figure 7, the method begins with receipt of a call requesting voice services in step 1210. To help ensure system integrity and to prevent unauthorized access, a call request is authenticated in step 1220. According to one embodiment, each incoming caller is automatically prompted to enter a login identifier and a PIN. According to another embodiment, an automatic number identification system is used, in addition to a login identifier and PIN system, to determine whether or not the user is calling from an authorized device. According to another embodiment, speaker recognition technology is utilized to identify a caller. According to this embodiment, voice prints for each user of the voice service system are stored as identifiers. When an inbound call is connected, pattern matching techniques are used verify the user's speech against the previously stored voice prints. Other security measures are possible.

In step 1230, a voice page is located. As explained above, a telecast of a voice service is driven by an active voice page. Accordingly, a user calling in to access voice services locates the desired active voice page. According to one embodiment, the user is automatically placed into an active voice page of a voice service that the user missed. That is, the system chooses an active voice page that it was unable to deliver. According to this embodiment, when a call is undeliverable (e.g., when an answering machine picks up), the active voice page for that call is placed in memory in a "voice site" table or as an

active voice page on a web site and addressed using the user's identification. When the user calls in to retrieve the voice service, after the user logs in, the table or web site will be searched for an active voice page that corresponds to their identification. If such a page exists, it is executed by the call server.

Other possibilities exist for accessing active voice pages through inbound calling. According to another embodiment, the system maintains a log of all voice services sent and provides an inbound user an option to select one of their previous voice services. According to another embodiment, an inbound caller is automatically placed into an active voice page that presents the user with an option to select one of that user's most frequently used services. According to still another embodiment, the user is allowed to search for past active voice pages by date or content. For example, the user may be prompted to enter a date on or near which the desired voice page was executed. According to another embodiment, the user may use the telephone keys to enter a search term and search the content of any previously executed active voice page that they are authorized to access or that is not secure.

Once an active voice page is located, the user navigates through the active voice page in step 1240. As described above, a user navigates through an active voice by exercising options, responding to prompts and otherwise entering input to the system. An inbound calling system would thus have access to the full functionality of the voice service system described in conjunction with Figures 1-6.

Figure 8 depicts a block diagram of a call server 18a that enables integrated inbound and outbound calling. In addition to the modules depicted in call server 18 of

Figure 3, call server 18a comprises call receiver module 1817, security module 1818 and search module 1819. Moreover, in the system for permitting inbound and outbound calling, call database 1811 has been replaced with an enhanced call database 1811a.

In order to receive inbound calls, call server 18a comprises call receiver module 1817. Although, call server 18 discussed above contains hardware permitting reception of calls as well as transmission of calls, it is not set up to receive calls. Call receiver module 1817 enables call server 18a to receive calls and routes the incoming calls to security module 1818. According to one embodiment, call receiver module comprises a software component designed to configure call server 18a to receive calls. Other embodiments are possible.

Received calls are forwarded to security module 1818 for authentication. According to one embodiment discussed above, incoming calls are authenticated using login I.D.'s and passwords. According to another embodiment, automatic number identification software is used to identify and authenticate callers. According to another embodiment, speech recognition and pattern matching techniques are used to identify a caller.

Authenticated calls may search for an active voice page using search module 1819. According to one embodiment, search module 1819 comprises a search engine designed specifically to search active voice pages. According to one embodiment discussed above, active voice pages utilize an XML-based language and search module 1819 comprises an XML-based search engine. According to another embodiment, search

module 1819 comprises a SQL engine designed to make queries against a relational or other type of database.

The active voice pages that are being search are stored in enhanced call database 1811a. In addition to its facilities to queue and log calls, enhanced call database 1811 includes facilities to catalog active voice pages. According to one embodiment, enhanced call database comprises a relational or other type of database. According to this embodiment, enhanced call database is used to store and categorize active voice pages and corresponding parameters, such as expiration dates for active voice pages. Other storage facilities are possible.

Various features and functions of the present invention extend the capabilities of previously known information delivery systems. One such system is MicroStrategy's Broadcaster version 5.6. The features and functions of the present invention are usable in conjunction with Broadcaster and other information delivery systems or alone. Other products may be used with the various features and functions of the invention including, but not limited to, MicroStrategy's known product suite.

Other embodiments and uses of the invention will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. The specification and examples should be considered exemplary only. The scope of the invention is only limited by the claims appended hereto.